IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A system, comprising:

a non volatile memory in which is stored an error handling routine, said error handling routine to permit a computer system to continue operating when an error is detected; and

a plurality of processors included in the computer system, wherein each processor of the plurality is capable of accessing the error handling routine on detecting an error and signaling ~~the~~ remaining processors of the plurality of processors to enter a rendezvous state.

2. (Original) The system of claim 1, wherein the error is only correctable by entering the rendezvous state.

3. (Original) The system of claim 1, further comprising a processor abstraction layer located in the non volatile memory, wherein the processor abstraction layer includes the error handling routine.

4. (Original) The system of claim 1, further comprising a system abstraction layer located in the non volatile memory, wherein the system abstraction layer includes the error handling routine.

5. (Original) A system, comprising:

a non volatile memory to store an error handling routine and an idle routine, said error handling routine to permit a computer system to continue operating when an error is detected;

a plurality of slave processors to execute the idle routine, wherein the plurality of slave processors are included in the computer system; and

a monarch processor included in the computer system, the monarch processor being capable of executing the error handling routine to correct the error.

6. (Original) The system of claim 5, wherein the error handling routine is included in a system abstraction layer.

7. (Original) The system of claim 5, wherein the monarch processor is capable of sending a wake up signal to the plurality of slave processors to exit the rendezvous state.

8. (Original) A system, comprising:

a plurality of processors including a monarch processor;

a processor abstraction layer coupled to the plurality of processors;

a system abstraction layer coupled to the processor abstraction layer;

an operating system layer coupled to the system abstraction layer; and

an interrupt signaling mechanism coupled to the processor abstraction layer, the system abstraction layer, and the operating system layer to initiate a rendezvous state and to end the rendezvous state, said rendezvous state being a state where all of the plurality of processors but the monarch processor are idle.

9. (Original) The system of claim 8, further comprising a system memory and a non volatile memory, wherein the operating system layer is located in the system memory and capable of being executed by the plurality of processors, the processor abstraction layer is located in the non volatile memory and is capable of being executed by the plurality of processors, and the system abstraction layer is located in the non volatile memory and is capable of being executed by the plurality of processors.

10. (Original) The system of claim 8, wherein the monarch processor is capable of executing an error handling routine included in the system abstraction layer upon initiation of the rendezvous state.

11. (Original) The system of claim 8, wherein the processor abstraction layer includes a functional module for error handling.

12. (Original) A system, comprising:

a plurality of processors;

a processor abstraction layer located in a non volatile memory coupled to the plurality of processors;

a system abstraction layer located in the non volatile memory; and

an operating system layer located in a system memory coupled to the plurality of processors to signal all but one of the plurality of processors to end a rendezvous state upon receiving a signal that error handling is completed, said rendezvous state being a state wherein all but the one of said plurality of processors are idle.

13. (Original) The system of claim 12, wherein the signal from the operating system to end the rendezvous state is an interrupt.

14. (Original) The system of claim 12, wherein the processor abstraction layer is capable of sending a signal to the system abstraction layer to enter the rendezvous state and performing error handling upon entering the rendezvous state.

15. (Original) A method, comprising:

detecting an error by one processor included in a multiple processor system;

entering a rendezvous state in which all processors but the one processor included in the multiple processor system are idle;

correcting the error using the one processor; and

resuming normal operation.

16. (Original) The method of claim 15, wherein entering the rendezvous state comprises:

requesting a plurality of processors included in the multiple processor system to enter an idle state; and

waiting until the plurality of processors have entered the idles state.

17. (Original) The method of claim 15, further comprising:

determining if the error is a severe error; and

only upon determining that the error is a severe error, entering the rendezvous state.

18. (Original) A method, comprising:

attempting to correct an error by a detecting processor included in a multiple processor system;

on failure, executing firmware code operatively coupled to all the processors included in the multiple processor system to correct the error; and

on failure, entering a rendezvous state to correct the error, said rendezvous state being a state where all but one of the processors included in the multiple processor system are idle.

19. (Original) The method of claim 18, wherein entering a rendezvous state to correct the error comprises:

selecting a monarch processor from the processors included in the multiple processor system;

signaling slave processors included in the multiple processor system to execute a spin loop; and

correcting the error by the monarch processor.

20. (Original) The method of claim 19, wherein correcting the error by the monarch processor includes executing routines in a processor abstraction layer, a system abstraction layer, and an operating system.

21. (Original) A method, comprising:

attempting to correct an error by a processor included in a plurality of processors;

accessing a routine in a first firmware layer to correct the error;

selecting a monarch processor included in the plurality of processors;

executing a spin loop routine in a second firmware layer by the plurality of processors except the monarch processor;

accessing a routine in the second firmware layer to correct the error; and

resuming normal operation by the plurality of processors.

22. (Original) The method of claim 21, wherein selecting a monarch processor comprises determining which processor included in the plurality of processors can best correct the error.

23. (Original) The method of claim 21, further comprising:

determining whether the error is severe.

24. (Original) An article comprising a machine-accessible medium having associated data, wherein the data, when accessed, results in a machine performing:

attempting to correct an error by a detecting processor included in a multiple processor system;

on failure, executing firmware code operatively coupled to all the processors included in the multiple processor system to correct the error; and

on failure, entering a rendezvous state to correct the error, said rendezvous state being a state where all but one of the processors included in the multiple processor system are idle.

25. (Original) The article of claim 24, wherein the rendezvous state is a state wherein all but the one of the processors included in the multiple processor system are executing a spin loop.

26. (Original) The article of claim 24, wherein the data, when accessed, results in the machine performing:

informing a processor abstraction layer when all but the one of the processors included in the multiple processor system have entered the idle state.